UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUEBEC, H3C 3J7

CRT 593

EFFICIENT IMPLEMENTATION OF HEURISTICS FOR THE CONTINUOUS NETWORK DESIGN PROBLEM

by

Patrice Marcotte[1,2]
and
Gérard Marquis[2]

[1] Collège militaire royal de Saint-Jean, Saint-Jean sur Richelieu, Québec, JOJ 1RO, Canada.

[2] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale A, Montréal, Québec, H3C 3J7, Canada.

This research was supported in part by a grant from the N.S.E.R.C. (A5789) and the PRU (FUHBP) from the Department of National Defence.

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal
Centre de recherche sur les transports - Publication #593
August 1988

Dépôt légal - 3e trimestre 1988

ABSTRACT

In this paper we present an efficient implementation of heuristic procedures for solving the continuous network design problem where network users behave according to Wardrop's first principle of traffic equilibrium. Numerical results involving a "standard" benchmark problem are given. Also it is shown that the cost mapping arising in the Iterative-Optimization-Assignment algorithm is integrable if and only if the volume-delay function is of the BPR type, or of some logarithmic form.

Key words : optimization, networks, transportation.

- - - - - -

RÉSUMÉ

Dans cet article nous montrons comment implanter efficacement des algorithmes de résolution d'un problème de design de réseau avec comportement "égoïste" des usagers. Des tests numériques sur un réseau de taille moyenne démontrent l'efficacité des procédures proposées. Enfin nous prouvons que l'équilibre de Nash correspondant au problème de design peut être résolu comme un problème d'optimisation convexe dans le cas où les fonctions de congestion sont de type BPR ou de type logarithmique.

Mots-clés : optimisation, réseaux, transport.

Introduction.

In this paper we address a network design problem involving design parameters, referred to as capacities, and private vehicles flows. It is assumed that the network is subject to congestion and that users behave according to Wardrop's first principle of traffic equilibrium, i.e. that, at equilibrium, no user can decrease his/her travel time by a unilateral change of route. This formulation is similar to that used in Abdulaal and Leblanc [1], Friesz, Suwansirikul and Tobin [3] or Marcotte [8] [9]. In the latter papers, several efficient heuristics were compared on networks where all arcs were subject to improvements and had zero initial capacities. In this paper we show that those heuristics perform equally well on networks where only a subset of the arcs are considered for improvement, and might have positive initial capacities. The main difference with the previous analysis results from the fact that the capacity-flow relationship is not as simple, and might even be nondifferentiable. The paper is organized as follows: First we give a mathematical formulation of the problem, then we describe the algorithms and their implementation. Next, numerical results on a small network and the benchmark "Sioux Falls" network are presented. Finally we give necessary and sufficient conditions under which the popular Iterative-Optimization-Assignment algorithm yields implicitly a descent direction respective to a related convex objective function.

Notation. Problem formulation.

Consider a transportation network $G = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is a node set and $\mathcal{A}$ an arc set. To each pair of nodes $k = (i,j) \in \mathcal{N} \times \mathcal{N}$ is associated a demand δ^k. A multicommodity flow vector $F = \{f_a^k\}_{a \in \mathcal{A}, k \in \mathcal{N} \times \mathcal{N}}$ is feasible if it is nonnegative and satisfies the flow conservation equations:

$$\sum_{j \in \mathcal{N}} f^k_{a=(i,j)} - \sum_{j \in \mathcal{N}} f^k_{a=(j,i)} = \begin{cases} -\delta^k, & \text{if } k = (i,l); \\ \delta^k, & \text{if } k = (l,i); \\ 0 & \text{otherwise.} \end{cases}$$

The polyhedron of feasible flow vectors F will be denoted by the letter Φ.

We will make use of the following notation:

$f_a = \sum_{k \in \mathcal{N} \times \mathcal{N}} f_a^k$: total flow on arc a.
$f = \{f_a\}_{a \in \mathcal{A}}$: total flow vector.

y_a: capacity of arc a.

$y = \{y_a\}_{a \in A}$: capacity vector.

$C_a(f, y)$: congestion (delay) function on arc a.

$C = \{C_a\}_{a \in A}$: congestion vector.

For a given capacity vector y, a multicommodity flow vector $F(y)$, together with the total flow vector $f(y)$, is in equilibrium (see Smith [12] or Dafermos [2]) if it satisfies the variational inequality:

$$\begin{aligned} &\langle f(y) - \overline{f}, C(f(y), y) \rangle \leq 0 \\ &\forall F \in \Phi \text{ where } f(y) \text{ corresponds to } F(y). \end{aligned} \tag{VIP}$$

The continuous network design problem (NDP) consists in choosing a capacity vector y in a feasible set Y which is optimal for the generalized bilevel program:

$$\begin{aligned} \min_{y \in Y} \langle f^t(y), C(f(y), y) \rangle \quad &\{\text{Transportation cost}\} \\ +\lambda g(y) \quad &\{\text{Investment cost}\} \end{aligned} \tag{NDP}$$

where g represents the cost associated with the capacity vector y, $f(y)$ the equilibrium total flow corresponding to y and λ a time/money conversion factor. The problem NDP is well defined if the total flow f corresponding to the F-solution to VIP is unique. A sufficient condition for uniqueness is that the cost function $C(f, y)$ be strictly monotone in f.

In the paper we will make the following assumptions:

Assumption 1. C_a is a strictly increasing function of the ratio f_a/y_a.

Assumption 2. g is separable, i.e. $g(y) = \sum_{a \in A} g_a(y_a)$ where each function g_a is nonnegative and increasing.

According to the previous assumptions, NDP can be written as the bilevel programming problem:

$$\begin{aligned} &\min_{f \in \Phi, y \in Y} \sum_{a \in A} f_a C_a(f_a/y_a) + \lambda g_a(y_a) \qquad \{\text{upper level}\} \\ &\text{s.t. } y \in \arg\min_{f \in \Phi} \sum_{a \in A} \int_0^{f_a} C_a(t/y_a)\, dt \quad \{\text{lower level}\}. \end{aligned} \tag{BLP}$$

It is also possible to express NDP as a mathematical program with nonlinear non-convex but finitely many constraints (see Marcotte [9]).

Heuristics for BLP. NDP is theoretically a difficult problem. Indeed its objective is nonconvex and nondifferentiable. Furthermore each evaluation of the reaction function $f(y)$ requires the exact solution of a fixed demand traffic assignment problem. In this paper we propose to utilize two members of the family of algorithms previously introduced by Marcotte [8], and described below.

1. Consider the normative mathematical program:

$$\min_{f\in\Phi, y\in Y} \sum_{a\in A} f_a C_a(f_a/y_a) + \lambda g_a(y_a). \qquad \text{(NORM)}$$

This is a convex program whose optimal value obviously provides a lower bound for BLP. Let $(f^{\text{NORM}}, y^{\text{NORM}})$ be optimal for NORM. We can then extract the suboptimal solution:

$$(f^0, y^0) \stackrel{\text{def}}{=} (f^{\text{NORM}}, y^{\text{NORM}})$$

by performing a traffic assignment on the network. A sufficient condition under which (f^0, y^0) is optimal for BLP is given in Marcotte [8].

2. Consider the convex mathematical program:

$$\min_{F\in\Phi, y\in Y} \sum_{a\in A} \int_0^{f_a} C_a(t/y_a)\, dt + \xi\lambda g_a(y_a) \qquad \text{(H5)}$$

whose solution is (f^5, y^5). By construction of H5 we have that $f^5 = f(y^5)$, i.e. that f^5 is the equilibrium total flow corresponding to the capacity vector y^5. The reason for the code name H5 is historical (see [8]).

An efficient implementation of NORM and H5.

We will concentrate on the solution of H5. The numerical solution of H0 is similar, with the exception of an added traffic assignment that must be performed in order to restore feasibility.

The constraint $y \in Y$ on the capacity vector will in general assume a simple form, such as $y_a \geq 0$. We make:

Assumption 3. $Y = \{y | y_a \geq 0, a \in \mathcal{A}1, y_a = 0, a \in \mathcal{A}2\}$ where $\mathcal{A}1$ represents the set of arcs subject to capacity improvement and $\mathcal{A}2 = \mathcal{A} - \mathcal{A}1$.

It is then possible, for fixed total flow vector f, to solve optimally for y as a function of f. More precisely one has, for each $a \in \mathcal{A}1$, to solve the equation:

$$\frac{\partial}{\partial y_a}\left[\int_0^{f_a} C_a(t/y_a)\,dt + \xi\lambda g_a(y_a)\right] = 0$$

or

$$\phi_a(f_a, y_a) \stackrel{\text{def}}{=} \int_0^{f_a} \frac{t}{y_a^2} C_a'(t/y_a)\,dt + \xi\lambda g_a'(y_a) = 0. \tag{1}$$

Let $\overline{y}_a(f_a)$ be the unique solution to the equation:

$$\frac{\partial}{\partial y_a}\phi_a(f_a, y_a) = 0.$$

Then we set:

$$y_a(f_a) = \max\{0, \overline{y}_a(f_a)\}. \tag{2}$$

Remark. In (2) it is implicit that the capacity of an arc can actually be decreased below its present value at no extra cost. This could actually improve the design in certain circumstances such as Braess paradox. □

After substituting expression (2) for y_a in H5 we obtain the network constrained convex program:

$$\min_{F \in \Phi} \sum_{a \in A} \int_0^{f_a} C_a(t/y_a(f_a)) + \xi\lambda g_a(y_a(f_a)) \tag{3}$$

that can be solved by any standard multicommodity convex flow algorithm, most of which are based on the Frank-Wolfe algorithm or some of its variants (PARTAN [4], simplicial decomposition [7], Fukushima's algorithm [5], etc.). The function $y_a(f_a)$ is generally nondifferentiable, due to the nondifferentiability of the function $g_a(y_a)$. This difficulty is best handled by approximating g_a by a smooth function. The most important case arises when g_a is linear:

$$g_a(y_a) = \max\{0, d_a(y_a - k_a)\}. \tag{4}$$

In this case we use, as approximating function, the branch of hyperbola whose equation is given by:

$$\frac{d_a}{2}\left[y_a - k_a + \sqrt{\epsilon^2 + (y_a - k_a)^2}\right] \tag{5}$$

where ϵ represent a small positive number and k_a the initial capacity of arc a (see figure 1). In the sequel it is assumed that such an approximation has been performed, and hence that $g_a(y_a)$ is a twice continuously differentiable function. It is to be noted that the nonnegativity constraint $y_a \geq 0$ is automatically enforced.

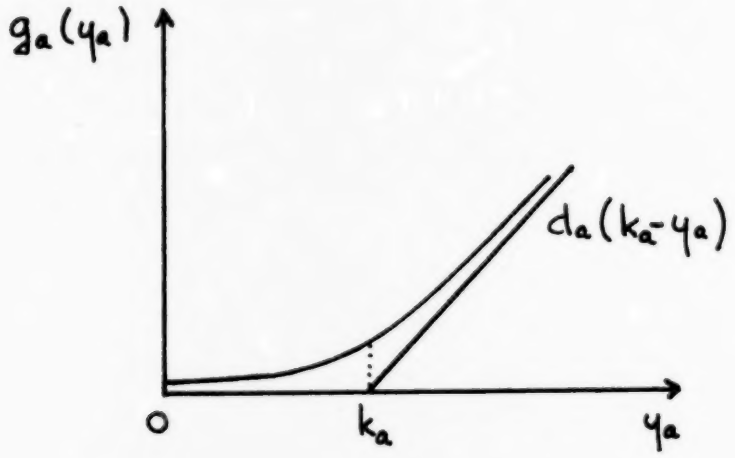

Figure 1. Linear investment function.

Remark. An implicit advantage of the approximation scheme is that, whenever given the choice, the algorithm will favor *low* versus *high* capacities. For instance, if some sort of Braess paradox phenomenon occurs, the algorithm could prevent it by lowering the capacities of arcs unused in the system-optimal solution. □

The minimization of (3) has been achieved using the linearization algorithm of Frank and Wolfe adapted to the traffic assignment problem (see Nguyen [11]). When not available in closed form, the function $y_a(f_a)$ was evaluated using the method of the false position ("regula falsi"). The first derivative of $y_a(f_a)$ was obtained by

implicit differentiation of ϕ_a:

$$\frac{\partial \phi_a}{\partial f_a}(f_a, y_a(f_a))y_a'(f_a) + \frac{\partial \phi_a}{\partial y_a}(f_a, y_a(f_a)) = 0. \tag{6}$$

One step of the algorithm is summarized below:

1. Find a descent direction: let

$$\overline{f} \in \arg \min_{F \in \Phi, y \in Y} \psi(f) \stackrel{\text{def}}{=} \sum_{a \in A} f_a \cdot \frac{\partial}{\partial x}\left[\int_0^x C_a(t/y_a(x))\,dt + \xi \lambda g_a(y_a(x))\right]|_{x=y_a^k}. \tag{7}$$

2. Perform linesearch (regula falsi or safeguarded Newton). Let

$$\alpha \in \arg \min_{\alpha \in [0,1]} \psi[f + \alpha(\overline{f} - f)].$$

The algorithm is stopped as soon as the gap function (see Hearn [6]) becomes less than a predetermined small value.

The computational burden of the algorithm consists mainly of two things:

1. Solving (7) by shortest path methods. This can be achieved in $O(Kn^2)$ or $O(Km \lg n)$ operations (where K denotes the number of origins or destinations, n the number of nodes in the network, and m the number of arcs) using Dijkstra's label setting algorithm.

2. Evaluating $y_a(f_a)$ and $y_a'(f_a)$ for $a \in A1$ in the linesearch. This implies $O(Lm_1)$ operations where L denotes the (usually small) number of regula falsi steps.

The overall one-step complexity is therefore $O(Kn^2 + Lm_1)$ or $O(Kn \lg n + Lm_1)$ if one implements Dijkstra's algorithm using a heap structure. The latter implementation is more efficient if m is of the order of n, which is usual for urban transportation networks. If the network is large and m_1 small, two things likely to occur in practice, then the term $Km \lg n$ dominates, and solving H5 is computationally comparable to performing a *single* traffic assignment on the network.

Numerical examples.

In the following test problems the congestion functions are of the BPR type, i.e.:

$$C_a(f_a, y_a) = \alpha_a + \beta_a (f_a/y_a)^p$$

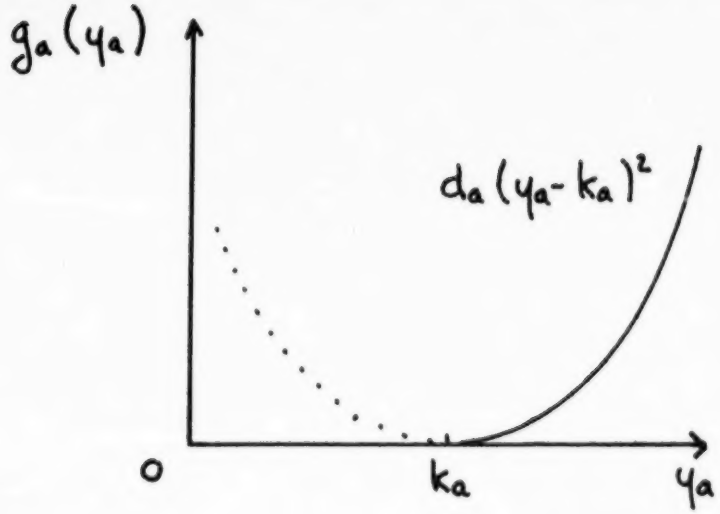

Figure 2. Quadratic investment function.

where α_a is usually referred to as the free flow travel time on arc a. It is worth noting that, in this particular case, heuristic H5 subsumes the Iterative-Optimization-Assignment algorithm (see [8] for a proof of this result), the latter corresponding to $\xi = 1/(p+1)$. In the appendix it is shown that the only functional forms for which the IOA actually solves (implicitly) a convex program are the BPR form and the logarithmic form.

In the test problems, investment functions are either linear or quadratic. As mentioned previously, linear investment functions are approximated by branches of hyperbola. In the quadratic case we have:

$$g_a(y_a) = \min\{0, d_a(y_a - k_a)^2\} \tag{8}$$

which is once but not twice continuously differentiable. However we can substitute to (8) the smooth function

$$g_a(y_a) = d_a(y_a - k_a)^2$$

without modifying the optimal solution, since values of y_a less than k_a are dominated by $y_a = 0$ in the solution of (7) (see figure 2).

The first example, taken from [3], consists of the network shown in figure 2. Demand is equal to D from node 1 to node 6, and $2D$ from node 6 to node 1. Coefficients α_a, β_a, k_a and l_a (the investment functions are linear) are given in [3]; λ has been set to 1.

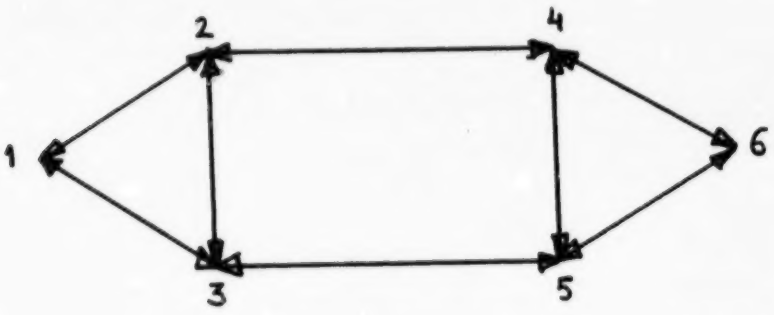

Figure 3. Network for the first example.

D	EDO	IOA	NORM	H5 $\xi = .2$	H5 $\xi = \xi^{opt}$
2.5	91.8 [57]	100.4	91.07	101.2	93.5 ($\xi = .45$)
5	200.89 [71]	214.3	199.7	213.3	213.3 ($\xi = .3$)
10	540.1 [68]	557.8	534.8	559.6	558.7 ($\xi = .22$)

Table 1. Results for the first example.

The objective function values in table 1 differ slightly from those reported in [3]. Our values have been computed by performing a very accurate assignment, using the y-values given in [3]. The precision with which the assignment has been performed might also explain the (small) discrepancies observed between the IOA result given in [3] and the results obtained by setting ξ to .2 in H5. These results should theoretically be identical.

Algorithms NORM and H5 were stopped after 50 iterations. At this point the relative gap value was less than .003. On this small example the best results were consistantly obtained by NORM. It is to be noted that the number of Frank-Wolfe steps could be lowered significantly without greatly affecting the quality of the solution.

The second example, first considered by Abdulaal and Leblanc [1], was also considered in [3] where the complete data can be found. It is the network of the city of Sioux Falls and is composed of 24 nodes, 76 arcs and the origine-destination matrix is full. A subset of 10 arcs are subject to improvements. Investment functions are quadratic. An absolute lower bound of 79.53 has been obtained by solving the system-optimal problem. Numerical results are summarized in table 2 below.

	EDO	HJ1	HJ2	NORM	H5 $\xi = .25$	H5 $\xi = .2$
objec.	83.68	82.32	82.06	81.61	81.78	81.84
iter.	89	247	147	56	54	53

Table 2. Results for the Sioux Falls network.

The names HJ1 and HJ2 refer to two different settings for the upper bounds of the improvement vectors (see [3]). The best solution was obtained by heuristic NORM with an objective value very close to the lower bound. The improvement vector $(y - k)$'s entries were: (5.48, 2.08, 5.49, 2.09, 2.85, 2.85, 4.23, 4.27, 4.24, 4.27) to be contrasted with the solution (4.59, 1.52, 5.45, 2.33, 1.27, 2.33, .41, 4.59, 2.71, 2.71) reported in [3]. For heuristic NORM, about one half of the iterations where used to solve the normative problem, and the remaining half to obtain a very accurate (.003 relative gap value) feasible (equilibrium) solution.

To assess the efficiency of algorithm H5, it was decided to check the quality of the solution obtained when stopping the procedure prematurily. For a given iteration count, the objective has been computed to within .003 relative gap. The results are given in table 3, for H5 ($\xi = .25$).

iter	5	10	15	20	30	40
y	4.41	4.75	4.73	4.83	4.82	4.85
	1.90	1.50	1.39	1.47	1.36	1.33
	4.36	4.60	4.72	4.82	4.87	4.85
	1.45	1.58	1.40	1.44	1.36	1.32
	2.62	2.30	2.48	2.37	2.42	2.39
	2.27	2.47	2.35	2.40	2.40	2.41
	4.73	4.36	4.28	4.21	4.22	4.20
	4.23	4.06	4.11	4.05	4.02	4.23
	4.75	4.47	4.38	4.27	4.21	4.23
	4.27	4.07	4.04	4.04	3.98	3.96
objec	82.04	81.81	81.79	81.79	81.81	81.80

Table 3. Evolution of algorithm H5 ($\xi = .25$).

From the above results it is clear that a very good solution can be obtained with very little effort (5 to 10 iterations). Actually the quality of the solution decreases when the number of Frank-Wolfe iterations increases from 20 to 30. This was to be expected since the problem solved by H5 is related but distinct from the original bilevel program. Similar results were obtained for heuristic NORM. Finally table 4 illustrates the behaviour of the objective as a function of the parameter ξ. The resulting curve is convex-shaped. this feature could be exploited if one is looking for the optimal value for ξ.

ξ: .15	.20	.25	.30	.35	.40	.45	1.0
obj. : 82.38	81.84	81.78	81.95	82.2	82.59	82.83	86.18

Table 4. Evolution of the objective as a function of ξ.

Remark. The algorithm was also tested on a five arc network (Braess paradox network). Heuristic H5, with ξ set to .2, converged in three iterations to the optimal solution obtained using the MINOS nonlinear programming code, and reported in [3]. On this same example, slightly suboptimal results were obtained by EDO and Hooke-Jeeves in a number of iterations ranging from 24 to 38. □

Conclusion.

In this paper we showed that algorithms initially designed for solving NDP where all arcs are subject to improvement and have zero initial capacities could also solve very efficiently NDPs under more general conditions. Efficiency of the proposed methodology relies on deriving, in explicit or implicit form, the functional relationship relating optimal capacities to flows, thus reducing the initial problem to a convex traffic assignment problem. The heuristics are robust and easy to implement. In particular no initial lower or upper bouds on the design variables are required. Indeed the number of Frank-Wolfe steps required to obtain a good solution is so low that, facing a difficult problem, one should solve NDP using NORM and H5 with several different values for the parameter ξ.

REFERENCES

[1] Abdulaal, M. and LeBlanc, L. J. (1979), "Continuous equilibrium design models", Transportation research 13B, 19–32.

[2] Dafermos, S. C. (1980) "Traffic equilibrium and variational inequalities", Transportation Science 14, 42–54.

[3] Friesz, T. L., Suwansirikul,C. and Tobin, R. L. (1987), "Equilibrium Decomposed Optimization: A heuristic for the continous equilibrium network design problem", Transportation Science 21 (4), 254–263.

[4] Florian, M., Guélat, J. and Spiess, H. (1987), "An efficient implementation of the PARTAN variation of the linear approximation method for the network equilibrium problem", NETWORKS 17, 3, 319–340.

[5] Fukushima, M. (1984), "A modified Frank-Wolfe algorithm for solving the traffic assignment problem", Transportation Research 18B, 169–177.

[6] Hearn, D. W. (1981), "The gap function of a convex program", Operations Research Letters 1, 67–71.

[7] Lawphongpanich, S. and Hearn, D. W. (1984), "Simplicial decomposition of the asymmetric traffic assignment problem", Transportation Research 18B, 123–133.

[8] Marcotte, P. (1983), "Network optimization with continuous control parameters", Transportation Science 17, 181–197.

[9] Marcotte, P. (1986), "Network design problem with congestion effects: A case of bilevel programming", Mathematical Programming 34(1) 142–162.

[10] Marcotte, P. and Guélat, J. (1988), "A modified Newton method for solving the asymmetric traffic equilibrium problem", Transportation Science 22(2) 112–124.

[11] Nguyen, S. (1974) "An algorithm for the traffic assignment problem", Transportation Science 8, 203–216.

[12] Smith, M. J. (1979), "The existence, uniqueness and stability of traffic equilibria", Transportation Research 13B, 295–304.

APPENDIX

INTEGRABILITY OF THE COST FUNCTION ARISING IN THE IOA ALGORITHM

The IOA algorithm consists of iteratively repeating the following two steps:

1. Optimization step.

$$\min_{y \in Y} \sum_{a \in A} f_a C_a(f_a/y_a) + \lambda g_a(y_a).$$

2. Assignment step.

$$\min_{f \in \Phi} \sum_{a \in A} \int_0^{f_a} C_a(t/y_a)\, dt.$$

The IOA algorithm can be viewed as a block Jacobi scheme for finding a solution (y^*, f^*) to the variational inequality:

$$\begin{aligned} &\sum_{a \in A} (y_a^* - y_a) \left[-\frac{f_a}{y_a}^2 C_a'(\frac{f_a^*}{y_a^*}) + \lambda g_a(y_a^*) \right] \leq 0 \\ &\sum_{a \in A} (f_a^* - f_a) C_a(\frac{f_a^*}{y_a^*}) \leq 0 \\ &\qquad \forall y \in Y, f \in \Phi. \end{aligned} \tag{9}$$

We want to show that the only functional forms for which the cost function of the variational inequality (9) is integrable, i.e. is the gradient of some function convex in both y and f, are the BPR form:

$$\alpha_a + \beta_a (f_a/y_a)^p \tag{10}$$

where p is a nonnegative constant identical for *all* arcs of the network, and the logarithmic form:

$$\alpha_a + \beta_a \ln (f_a/y_a).$$

A direct consequence of this result is that, if the delay functions are of the form (10), then the IOA converges from an arbitrary feasible starting point (y^0, f^0) to the Nash equilibrium solution corresponding to (9).

The cost function of (9) will be integrable if there exists functions $\phi_a(f_a, y_a)$ satisfying the system of partial differe ıtial equations:

$$\begin{aligned} \frac{\partial \phi_a}{\partial y_a} &= -\left(\frac{f_a}{y_a}\right)^2 C_a'\left(\frac{f_a}{y_a}\right) + \lambda g_a'(y_a) \\ \frac{\partial \phi_a}{\partial f_a} &= \gamma C_a\left(\frac{f_a}{y_a}\right) \end{aligned} \tag{11}$$

where γ is an arbitrary positive constant. Indeed the solution set to (9) remains unchanged, and the IOA iterates are identical, if the second inequality is premultiplied by a positive constant.

To simplify the notation, we drop the separable term $g_a'(y_a)$ and the arc indices from (11) and obtain the system:

$$\begin{aligned} \frac{\partial \phi}{\partial y} &= -\left(\frac{f}{y}\right)^2 C'\left(\frac{f}{y}\right) \\ \frac{\partial \phi}{\partial f} &= \gamma C\left(\frac{f}{y}\right). \end{aligned} \tag{12}$$

Let: $h(x) \stackrel{\text{def}}{=} \int_0^x C(t)\,dt$. We have:

$$\begin{aligned} \phi(f, y) &= \int_0^f C\left(\frac{t}{y}\right) dt \\ &= y\, h\left(\frac{f}{y}\right) + K(y). \end{aligned}$$

Hence:

$$\begin{aligned} \frac{\partial \phi}{\partial y} &= h\left(\frac{f}{y}\right) - \frac{f}{y} h'\left(\frac{f}{y}\right) + K'(y) \\ &= -\gamma\left(\frac{f}{y}\right)^2 h''\left(\frac{f}{y}\right). \end{aligned}$$

From the above identity we have $K'(y) = 0$. Also, after setting $x = f/y$ we obtain the second-order linear homogeneous differential equation (with non constant coefficients):

$$\gamma x^2 h''(x) - x h'(x) + h(x) = 0 \tag{13}$$

which admits the two linearly independant solutions:

$$\begin{aligned} h_1(x) &= x \\ h_2(x) &= x^{1/\gamma}. \end{aligned}$$

For $\gamma \neq 1$, the general solution to (13) is:

$$h(x) = \alpha x + \beta x^{1/\gamma}$$

and we have:

$$C(x) = h'(x) = \alpha + \beta x^{1/\gamma - 1}$$

which is of the form (10) with $p = \frac{1}{\gamma} - 1$. Also p must be nonnegative since $C_a(f_a/y_a)$ has to be an increasing function of f_a over the nonnegative axis.

If $\gamma = 1$ then the general solution to (13) is:

$$h(x) = \alpha x + \beta x \ln x$$

which is of the required logarithmic form. QED